

*REMARKS*

In response to the Official Action dated April 6, 2004, Applicants request reconsideration. In this Response, no claims are added, canceled, or amended.

The Official Action rejected claims 1, 2, 4-12, 15, and 16 as unpatentable over Carlson et al. (US Patent 5,623,592, hereinafter Carlson) in view of Takizawa et al. (US Patent 5,565,748, hereinafter Takizawa). The rejection of claim 16 is inferred from the comments at page 8 of the Official Action. The rejection is not mentioned elsewhere. That rejection is respectfully traversed as to claims 1, 2, 4-12, 15, and 16.

An interview between Examiner Pham and Applicants' representative was conducted on June 17, 2004. The claim limitations were discussed with reference to Carlson and Takizawa. No agreement was reached.

The combination of Carlson and Takizawa fails to teach or suggest a software module, wherein the software module is *automatically linked to the development means* based on information stored on the object, as alleged by the Official Action. The Official Action asserts that Takizawa teaches this limitation at column 9, lines 49-55. Applicants respectfully submit that, in reading the claims on the description of Takizawa, the Official Action has confused the development means with the program being developed. Takizawa teaches that software modules are automatically combined to create an emulator program (column 9, lines 52-55 of Takizawa). Takizawa does not teach that software modules are automatically linked to the *development means* used by the operator to selectively combine those modules that are then used to automatically create the emulator program. That is, the present invention provides the advantage that software modules used in creating a program in a development means are automatically linked, based on information from the object, to that development means, and thus available for use by the operator. Takizawa teaches only that the software modules are provided in the development means, and not how they are provided (e.g. automatically).

Furthermore, in the present invention, the module is automatically linked to the development means *based on information stored on the object*, such as a globally unique identifier (GUID). Takizawa teaches or suggests nothing of the sort. In Takizawa, software modules are combined into an emulator program based on selection by an operator (see column 9, lines 49-52 of Takizawa). Thus, the Official Action is erroneous in asserting that Takizawa teaches linking based on anything other than operator selection. Moreover, the combination of Takizawa and Carlson fails to teach or suggest information of any kind stored on the object. In Carlson, the *object to be controlled*, as asserted by the Official Action, is device 104, 106, 108, or 110 (see Figure 1 of Carlson). Carlson fails to teach that information is stored on these

objects. In Takizawa, the object to be controlled is a servo amplifier 2a-d in combination with a motor 3a-d (see Figure 1 of Takizawa). Takizawa also fails to teach that information is stored on these objects. Thus, the Official Action is erroneous in asserting that the combination of Carlson and Takizawa teaches or suggests storing information on the object to be controlled.

Still further, the combination of Carlson and Takizawa fails to teach or suggest a software module providing at least one of a description procedure used in said development means for describing a control process for said object, and an implementing procedure for implementing the control process developed in said development means for said object. The Official Action asserts that it would have been obvious to one of ordinary skill in the art to have at least a description procedure for the control process associated with icons. The Official Action gives as an example the description window that appears when a mouse cursor is moved over an icon. Applicants respectfully submit that the Official Action has confused the limitations of the claims. The description procedure describes a procedure for controlling an object, and is no way related to icons. The pop-up window cited by the Official Action does not describe how to control an object.

Regarding claim 15 specifically, the Official Action does not assert that the combination of Carlson and Takizawa teaches or suggests an implementing procedure for implementing a control process, or a subprocess for acquiring a global unique ID from a device. In fact, the combination suggests neither limitation. Applicants also note for the record that the mere display of an icon on a screen does not teach or suggest a procedure for displaying an icon. Accordingly, neither Carlson nor Takizawa teaches or suggests a software module including icon procedure for displaying an icon.

Moreover, there is no motivation to combine Carlson and Takizawa as suggested by the Official Action. The Official Action asserts that it would have been obvious to incorporate the software module of Takizawa with the system of Carlson because the Carlson process control would not be possible without the unique software module that interfaces between the controller and external devices. Applicants are confused by this assertion. It seems that the Official Action contends that the invention of Carlson cannot work without the invention of Takizawa, which is plainly not true. Furthermore, Carlson does not require a *unique* software module to interface between the controller and external devices. It is equally possible that a general purpose driver is used to control all of the devices. Accordingly, the Official Action has not set forth an advantage that is gained by the combination of Carlson and Takizawa. The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. *In re Mills*, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990)

Thus, there is no motivation to combine Carlson and Takizawa, and the combination of Carlson and Takizawa fails to teach or suggest all limitations of claims 1 and 15. Accordingly, *prima facie* obviousness has not been established and the rejection of those claims, as well as dependent claims 2-14 and 16, is erroneous and should be withdrawn.

Regarding claim 16, the Official Action alleges that Carlson teaches a software module providing an icon procedure for display of an icon. However, Carlson merely discloses that an icon bar is displayed (see column 6, lines 32-40 of Carlson). The mere fact that icons are displayed does not suggest a software module providing an icon procedure for displaying an icon, as recited in claim 16. Carlson makes no mention of a *software module* providing an *icon procedure*. Carlson simply cannot be relied upon to teach this limitation. Accordingly, the combination of Carlson and Takizawa fails to teach or suggest all limitations of claim 16, thus *prima facie* obviousness has not been established.

The Official Action rejected claims 3 and 13 as unpatentable over Carlson in view of Takizawa and further in view of Kodosky et al. (US Patent 6,173,438, hereinafter Kodosky). That rejection is respectfully traversed.

Kodosky fails to teach or suggest those limitations of amended claim 1 that are absent from the combination of Carlson and Takizawa, as previously discussed. Accordingly, the combination of Carlson, Takizawa, and Kodosky cannot teach or suggest claims 3 and 13, which inherit the limitations of amended claim 1.

Moreover, Applicants respectfully traversed this rejection in the Amendment of January 23, 2004, yet the present Official Action has provided no rebuttal to Applicants' arguments regarding these claims. "Where the applicant traverses any rejection, the examiner should, if he or she repeats the rejection, take note of the applicant's argument and answer the substance of it" (see MPEP 707.07(f)). The argument previously presented is repeated here for the Examiner's convenience.

The Official Action erroneously contends that Kodosky teaches a development means acquiring the software modules from the object. Kodosky plainly states that the host system transfers software drivers to the embedded system (see column 13, lines 52-55 of Kodosky). Thus, the development means (host LabVIEW, see Abstract) clearly does not need to acquire the software modules assigned to the embedded system from the embedded system, since the host LabVIEW transferred the software modules to the embedded system originally.

The Official Action rejected claim 14 as unpatentable over Carlson in view of Takizawa and further in view of Kang (US Patent 6,279,049). That rejection is respectfully traversed.

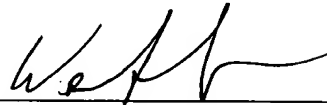
Kang fails to teach or suggest those limitations of amended claim 1 that are absent from the combination of Carlson and Takizawa, as previously discussed. Accordingly, the

In re Appln. of NAKAI et al.  
Application No. 09/845,361

combination of Carlson, Takizawa, and Kang cannot teach or suggest claim 14, which inherits the limitations of amended claim 1.

Reconsideration and allowance of claims 1-16 is earnestly solicited.

Respectfully submitted,



---

A. Wesley Ferrebee, Reg. No. 51,312  
LEYDIG, VOIT & MAYER  
700 Thirteenth Street, N.W., Suite 300  
Washington, DC 20005-3960  
(202) 737-6770 (telephone)  
(202) 737-6776 (facsimile)

Date: \_\_\_\_\_

AWF/tps

Amendment or ROA - Regular (Revised 9/03/03)

6/22/04